

# PERBANDINGAN GENETIC ALGORITHM, MULTIPLE ANT COLONY SYSTEM DAN TABU SEARCH UNTUK PENYELESAIAN VEHICLE ROUTING PROBLEM WITH TIME WINDOWS (VRPTW)

Basuki Rahmat<sup>1)</sup>

<sup>1)</sup> Jurusan Teknik Informatika, FTI, UPN "Veteran" Jawa Timur  
Email: basuki@upnjatim.ac.id

**Abstract.** *This research purpose is to compare the performance of GENETIC ALGORITHM (GA), MULTIPLE ANT COLONY SYSTEM (MACS) AND TABU SEARCH (TS) in mean to search optimal solution for Vehicle Routing Problem With Time Windows (VRPTW). The optimization value are total armada needed, travel time and travel distance. Simulation result for 50 customers problem show MACS give better performan than others.*

**Keywords :** GA, MACS, TS, VRPTW, PFH.

Permasalahan untuk meminimalkan biaya dan total waktu biasanya disebut dengan *Vehicle Routing Problem* (VRP). VRP adalah suatu nama umum yang diberikan kepada suatu kelas permasalahan utuh dengan satu set rute untuk armada angkut (*vehicle*) yang berasal dari satu atau lebih depot yang harus disebarkan untuk melayani beberapa pelanggan (*customer*). Tujuan dari VRP adalah untuk melayani pelanggan sesuai dengan permintaannya dengan meminimalkan biaya angkut yang dimulai dan berakhir di depot.

VRP adalah permasalahan kombinasi yang kompleks, yang dapat dilihat sebagai penggabungan dua permasalahan yang terkenal: *Traveling Salesman Problem* (TSP) dan *Bin Packing Problem* (BPP), yaitu diberikan suatu armada angkut dengan kapasitas seragam, suatu depot, dan beberapa pelanggan dengan permintaan (*demand*).

Kemudian permasalahannya adalah mencari suatu rute yang dilalui dengan keseluruhan biaya rute yang minimal yang dapat melayani semua permintaan pelanggan. Semua perjalanan dimulai dan diakhiri di depot dan rute perjalanan armada harus dirancang sehingga masing-masing pelanggan hanya dilayani sekali dan hanya oleh satu armada. VRP adalah NP (*Non Polynomial*)-Hardness dan oleh karena itu sulit untuk memecahkannya, maka untuk menyelesaikan masalah VRP diperlukan suatu algoritma pendekatan yang dapat membantu memecahkan masalah ini dengan mudah.

Sampai saat ini metode untuk memecahkan permasalahan VRP dikategorikan atas tiga generasi. Generasi pertama adalah metode *Exact Approachs* yaitu algoritma yang mencari dan menghitung tiap-tiap kemungkinan solusi sampai salah satu solusi terbaik tercapai. Kelemahan dari metode ini adalah waktu pencarian solusi yang lama bila jumlah pelanggan yang dicari besar. Generasi kedua adalah metode *Heuristic* yaitu algoritma yang membangun suatu eksplorasi pencarian terbatas dan menghasilkan solusi yang baik dengan waktu yang cepat. Generasi ketiga adalah metode *Metaheuristic* merupakan pengembangan dari metode *heuristic*, dengan cara melakukan eksplorasi yang lebih dalam/meningkatkan hasil solusi terbaik dari pendekatan *heuristic*, sehingga solusi yang dihasilkan jauh lebih baik dari pada hasil metode *heuristic*.

*Genetic algorithm* (GA), *Multiple Ant Colony System* (MACS) dan *Tabu Search* (TS) adalah tiga contoh algoritma dari metode *Metaheuristic*. Dalam penelitian ini ketiga metode ini akan dibandingkan untuk memilih yang terbaik dalam pencarian solusi kasus VRPTW.

## METODE

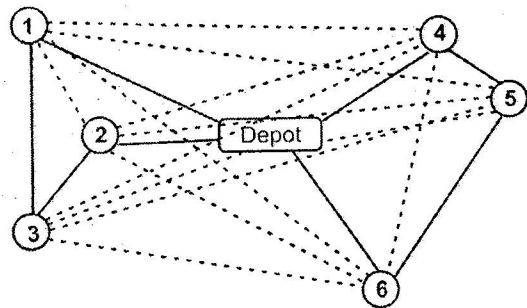
### Analisa Permasalahan

VRP merupakan permasalahan *Combinational Optimization*, yaitu terdapat sejumlah armada berpangkalan pada depot yang akan melayani sejumlah pelanggan, tiap-tiap

pelanggan punya permintaan dan setelah selesai melayani pelanggan maka armada akan kembali lagi ke depot.

Dari definisi di atas maka permasalahannya adalah bagaimana membuat rute perjalanan dari suatu armada dengan tujuan meminimalkan jumlah armada yang digunakan, kemudian meminimalkan total waktu perjalanan dan meminimalkan jarak perjalanan armada tersebut, dan tidak melanggar batasan yang telah ditentukan, yaitu batasan *time windows* dan batasan daya angkut armada.

Dari Gambar 1 dapat dilihat bahwa permasalahan VRPTW merupakan permasalahan yang kompleks, yaitu terdapat sejumlah pelanggan  $C$  untuk dilayani dengan sejumlah armada  $V$  dan satu buah depot. Tiap-tiap pelanggan terhubung dengan garis putus-putus yang menggambarkan jaringan jalan antar pelanggan, dari gambar dapat terlihat bahwa semakin besar pelanggan maka semakin besar pula kemungkinan solusi yang dicari.



Gambar 1. Permasalahan VRPTW

### Penyelesaian dengan Genetic Algorithm

Pada dasarnya GA menyelesaikan permasalahan dengan melakukan perkawinan silang antara kromosom terbaik yang dimilikinya, dengan harapan sifat-sifat baik dari kromosom-kromosom tersebut akan menurun pada kromosom keturunannya.

Inti dari GA adalah: (1) membuat populasi awal, (2) menghitung nilai fitness dari kromosom dalam populasi, (3) melakukan reproduksi untuk membentuk populasi baru, (4) menjadikan populasi baru menjadi populasi sekarang yang digunakan untuk generasi berikutnya.

Pada bagian inisialisasi tersebut terdapat generasi yang merupakan penanda berapa kali

generasi/proses reproduksi akan diberlakukan pada populasi. Kemudian pada awalnya kromosom dalam populasi adalah kosong, dan reproduksi tidak mungkin dilakukan dengan populasi yang kosong, sehingga dibutuhkan inisialisasi awal dari populasi. Pada penelitian ini populasi awal akan diisi dengan hasil solusi dari algoritma *Push Forward Insertion Heuristic* (PFIH). Metode ini merupakan salah satu metode *heuristic* biasa yang digunakan untuk menentukan solusi awal dari metode *metaheuristic*.

Setelah populasi awal terbentuk maka akan dihitung nilai *fitness* dari tiap-tiap kromosom dalam populasi. Nilai *fitness* ini berfungsi untuk mengindikasikan seberapa baik solusi dari kromosom tersebut berdasarkan fungsi tujuan, semakin besar nilai *fitness* maka semakin baik kromosom tersebut. Kemudian kromosom dengan nilai *fitness* tertinggi dijadikan solusi terbaik yang ditemukan  $S_{best}$ .

Proses dilanjutkan dengan reproduksi yang merupakan inti dari algoritma ini. Dari proses reproduksi ini akan dihasilkan populasi baru yang berisi kromosom-kromosom turunan yang baru atau yang sering disebut *offspring*. Setelah populasi baru terbentuk maka dihitung nilai *fitness* dari populasi baru tersebut. Lalu dari beberapa kromosom dengan nilai *fitness* terendah dalam populasi akan digantikan dengan sejumlah sama dengan kromosom dengan nilai *fitness* tertinggi dari populasi sekarang/lama.

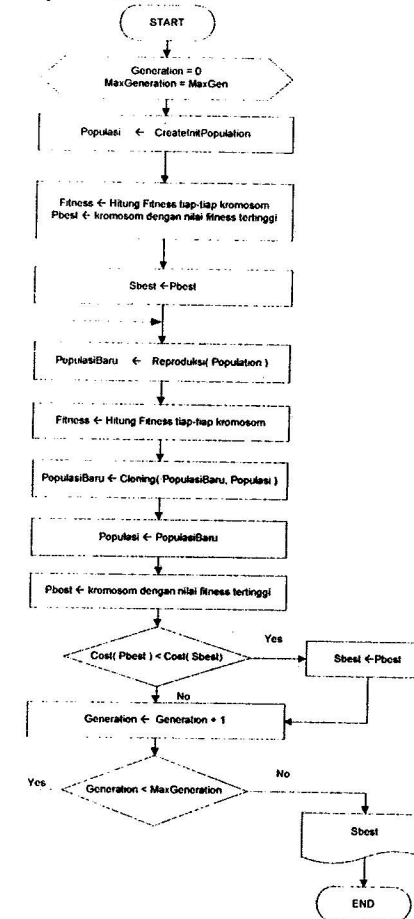
Setelah populasi baru terbentuk, kemudian populasi tersebut dijadikan populasi sekarang yang akan digunakan pada reproduksi berikutnya. Dan dari populasi tersebut dipilihlah kromosom dengan nilai *fitness* tertinggi dan kemudian dibandingkan dengan solusi  $S_{best}$ . Apabila ternyata lebih baik maka solusi  $S_{best}$  digantikan dengan solusi dari kromosom tersebut. Lalu proses reproduksi ini akan diulang-ulang sampai kriteria berhenti terpenuhi (yaitu jumlah generasi sama dengan jumlah generasi yang diharapkan). Dan hasil akhir dari solusi  $S_{best}$  adalah hasil solusi terbaik dari algoritma ini.

Diagram alir GA untuk penyelesaian kasus VRPTW dapat dilihat pada Gambar 2.

### Penyelesaian dengan MACS

Untuk menyelesaikan permasalahan VRPTW dengan algoritma MACS menggunakan dua buah *Ant Colony System*

(ACS) yang mempunyai tugas sendiri-sendiri, yaitu untuk koloni yang pertama digunakan untuk mencoba meminimalkan jumlah armada pengangkut yang digunakan untuk mendistribusikan permintaan dari depot ke pelanggan dan untuk koloni yang kedua digunakan untuk mencoba meminimalkan total *travel time* dari perjalanan yang telah dibuat oleh koloni pertama.



Gambar 2. Diagram alir GA

Berikut ini algoritma MACS untuk solusi kasus VRPTW.

### Algoritma : MACS-VRPTW

#### Procedure MACS-VRPTW

1. /\* Inisialisasi \*/

$S \leftarrow$  Buat Solusi awal dengan PFIH

$S_{best} \leftarrow S$

2. Repeat

$V \leftarrow \text{ActiveArmada}(S_{best})$

Activate ACS-VEI(  $V - 1$  )

Activate ACS-TIME(  $V$  )

While ACS-VEI dan ACS-TIME active

if (  $S_{best} < S$  ) then

$S_{best} \leftarrow S$

if (  $\text{ActiveArmada}(S_{best}) < V$  ) then

kill ACS-TIME dan ACS-VEI

end-while

until kriteria berhenti

Return  $S_{best}$

### End-procedure

ACS-TIME diaktifkan dengan tujuan mencoba untuk membangun solusi untuk meminimalkan total *travel time*/mengurangi *travel time* dari solusi  $S_{best}$  saat ini dengan menggunakan jumlah armada  $V$ . Apabila solusi yang dihasilkan ACS-TIME lebih baik dari solusi  $S_{best}$  maka solusi tersebut dijadikan solusi  $S_{best}$  sekarang dan tiap-tiap ada solusi yang lebih baik yang dihasilkan oleh kedua koloni tersebut maka ACS-VEI dan ACS-TIME dimatikan. Kemudian proses akan diulang lagi dan akan mengaktifkan koloni baru lagi. Proses tersebut akan dilakukan berulang-ulang sampai kriteria pemberhentian tercapai.

### Penyelesaian dengan Tabu Search

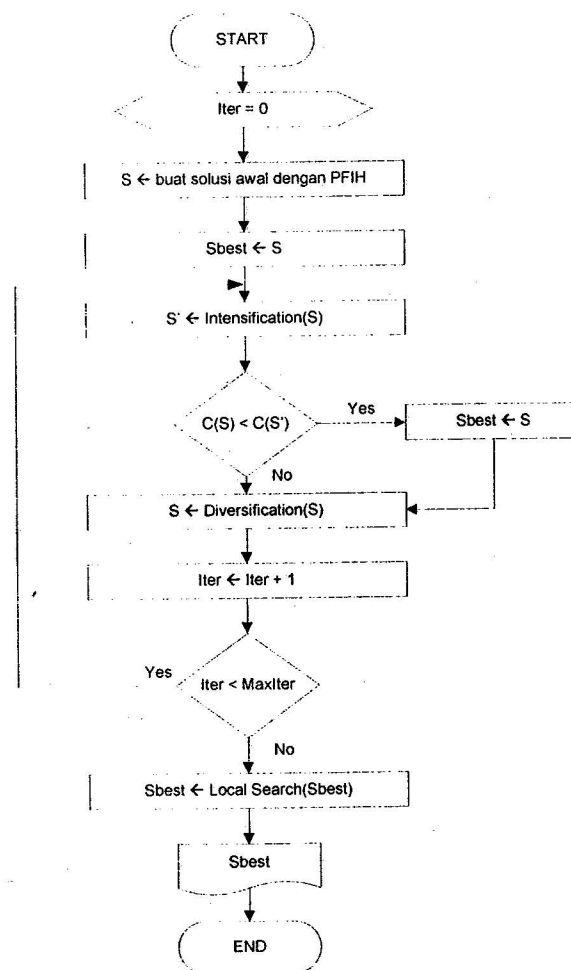
*Tabu Search* adalah teknik pencarian *neighbourhood* yang menggunakan memori yang disebut *tabu* untuk menghindari perulangan solusi, sehingga setiap perpindahan dari setiap iterasi adalah merupakan area pencarian solusi yang baru.

Pada Gambar 3 diperlihatkan urutan dari algoritma TS dalam menyelesaikan permasalahan. Pada awalnya dibutuhkan sebuah solusi awal  $S$  yang dibuat dengan algoritma PFIH dan solusi dari  $S$  dijadikan solusi terbaik  $S_{best}$  sementara.

Kemudian solusi  $S$  ditingkatkan kualitasnya dengan *intensification*, untuk proses dari *intensification* mirip dengan proses *local search*, tetapi perbedaannya dalam *intensification* akan diberi tambahan aturan *tabu list* dan aturan *aspiration criteria*. Hal ini diberikan untuk menjaga agar ruang solusi yang dicari adalah luas.

Setelah solusi  $S$  ditingkatkan kualitasnya dengan *intensification* kemudian solusi  $S$  dibandingkan dengan  $S_{best}$  saat ini, bila lebih baik maka ubah nilai solusi  $S_{best}$  sama dengan menjadi  $S$ . Setelah solusi  $S$  menjadi solusi yang lokal optima, maka diperlukan proses untuk keluar dari kondisi tersebut yang sering disebut proses *diversification* atau eksplorasi. Proses

*intensification* dan *diversification* akan dilakukan secara berulang-ulang sampai kriteria berhenti ditemukan. Setelah proses selesai maka untuk menjamin agar solusi yang dihasilkan adalah global optima maka pada akhir solusi  $S_{best}$  dioptimalkan lagi dengan *local search* dengan metode  $\lambda$ -Interchange.



Gambar 3. Diagram alir Tabu Search

## HASIL DAN PEMBAHASAN

### Uji Coba 50 Pelanggan

Permasalahan VRPTW dalam penelitian ini diujicoba dengan jumlah node pelanggan yang harus dilayani sebanyak 50 pelanggan, dengan jumlah permintaan seluruh pelanggan sebanyak 860 dan untuk kapasitas daya angkut satu armada adalah 200. Untuk detail dari tiap-tiap pelanggan dapat dilihat pada Tabel 1, yaitu untuk pelanggan ke 0 adalah menandakan depot.

Tabel 1. Permasalahan VRPTW dengan 50 pelanggan

No.	X	Y	D	E.T	L.T	S.T
0	40	50	0	0	1263	0
1	45	68	10	91	967	90
2	45	70	30	82	870	90
3	42	66	10	65	146	90
4	42	68	10	72	782	90
5	42	65	10	15	67	90
6	40	69	20	62	702	90
7	40	66	20	17	225	90
8	38	68	20	25	324	90
9	38	70	10	53	605	90
10	35	66	10	35	410	90
11	35	69	10	48	505	90
12	25	85	20	65	721	90
13	22	75	30	30	92	90
14	22	85	10	56	620	90
15	20	80	40	38	429	90
16	20	85	40	47	528	90
17	18	75	20	99	148	90
18	15	75	20	17	254	90
19	15	80	10	27	345	90
20	30	50	10	10	73	90
21	30	52	20	91	965	90
22	28	52	20	81	773	90
23	28	55	10	73	777	90
24	25	50	10	65	144	90
25	25	52	40	16	224	90
26	25	55	10	62	701	90
27	23	52	10	26	316	90
28	23	55	20	54	593	90

29	20	50	10	35	405	90
30	20	55	10	44	504	90
31	10	35	20	20	237	90
32	10	40	30	31	100	90
33	8	40	40	87	158	90
34	8	45	20	75	816	90
35	5	35	10	28	344	90
36	5	45	10	66	716	90
37	2	40	20	38	434	90
38	0	40	30	47	522	90
39	0	45	20	56	624	90
40	35	30	10	26	321	90
41	35	32	10	16	235	90
42	33	32	20	68	149	90
43	33	35	10	16	80	90
44	32	30	10	35	412	90
45	30	30	10	54	600	90
46	30	32	30	11	509	90
47	30	35	10	10	1127	90
48	28	30	10	63	693	90
49	28	35	10	10	1066	90
50	26	32	10	85	880	90

Keterangan:  
 X,Y: Position  
 D: Demand  
 E.T: early Time  
 L.T: Latest Time  
 S.T: Service Time

Dengan menggunakan data permasalahan di atas dilakukan percobaan membentuk solusi sebanyak 5 kali percobaan, didapatkan tabel hasil penyelesaian seperti pada Tabel 2.

Tabel 2. Penyelesaian Permasalahan 50 pelanggan

Percobaan		GA	MACS	TS
1	V T D	6 5420 552	5 4956 397	6 5037 437
2	V T D	6 5496 718	5 4956 397	6 5112 485
3	V T D	6 5246 545	5 4956 397	6 5231 533
4	V T D	6 5478 651	5 4956 397	6 5037 437
5	V T D	6 5403 572	5 4956 397	6 5109 483

Keterangan: V : jumlah armada  
 T : total waktu perjalanan (travel time)  
 D : total jarak perjalanan (travel distance)

Dari Tabel 2 terlihat adanya solusi yang diberi tulisan tebal menunjukkan bahwa solusi tersebut adalah solusi terbaik yang dapat ditemukan oleh masing-masing algoritma. Dari permasalahan 50 pelanggan menunjukkan bahwa kinerja dari algoritma MACS adalah yang terbaik, kemudian diikuti solusi dari algoritma TS dan terakhir adalah solusi algoritma GA.

Detail solusi dari tiap-tiap algoritma adalah sebagai berikut:

#### a. Solusi Genetic Algorithm

Pada Tabel 3. adalah hasil total dari pencarian solusi untuk tiap-tiap rute solusi dengan menggunakan algoritma GA. Tabel 4. adalah urutan perjalanan dari tiap-tiap armada untuk melayani pelanggan.

Tabel 3. Total solusi GA untuk 50 pelanggan

No Rute	Node	Demand	Travel Time	Travel Distance
1	4	50	404	44
2	7	150	749	92
3	8	120	835	49
4	11	200	1233	170
5	11	160	1049	59
6	9	180	975	131
Total	50	860	5246	545

Tabel 4. Detail solusi GA untuk 50 pelanggan

No Rute	Urutan Perjalanan
1	0 - 43 - 42 - 41 - 40 - 0
2	0 - 32 - 33 - 31 - 44 - 46 - 45 - 48 - 0
3	0 - 24 - 25 - 27 - 29 - 30 - 28 - 26 - 23 - 0
4	0 - 20 - 18 - 19 - 15 - 16 - 14 - 12 - 50 - 21 - 49 - 47 - 0
5	0 - 5 - 3 - 7 - 8 - 10 - 11 - 9 - 6 - 4 - 2 - 1 - 0
6	0 - 13 - 17 - 35 - 37 - 38 - 39 - 36 - 34 - 22 - 0

## b. Solusi Multiple Ant Colony System

Pada Tabel 5 adalah hasil total dari pencarian solusi untuk tiap-tiap rute solusi dan total penjumlahan nilai-nilai dari seluruh rute yang ditemukan oleh algoritma MACS. Tabel 6 adalah urutan pelanggan yang dikunjungi oleh tiap-tiap armada.

Tabel 5. Total solusi MACS untuk 50 pelanggan

No Rute	Node	Demand	Travel Time	Travel Distance
1	10	150	951	51
2	12	160	1233	94
3	11	160	1049	59
4	9	200	907	97
3	8	190	816	96
<b>Total</b>	<b>50</b>	<b>860</b>	<b>4956</b>	<b>397</b>

Tabel 6. Detail solusi MACS untuk 50 pelanggan

No Rute	Urutan Perjalanan
1	0 - 20 - 24 - 25 - 27 - 29 - 30 - 28 - 26 - 23 - 22 - 0
2	0 - 43 - 42 - 41 - 40 - 44 - 46 - 45 - 48 - 50 - 21 - 49 - 47 - 0
3	0 - 5 - 3 - 2 - 8 - 10 - 11 - 9 - 6 - 4 - 2 - 1 - 0
4	0 - 32 - 33 - 31 - 35 - 37 - 38 - 39 - 36 - 34 - 0
5	0 - 13 - 17 - 18 - 19 - 15 - 16 - 14 - 12 - 0

## c. Solusi Tabu Search

Pada Tabel 7 adalah hasil total dari pencarian solusi untuk tiap-tiap rute solusi menggunakan algoritma TS. Tabel 8 adalah urutan perjalanan dari tiap-tiap armada untuk melayani pelanggan.

Tabel 7. Total solusi TS untuk 50 pelanggan

No Rute	Node	Demand	Travel Time	Travel Distance
1	11	170	1041	51
2	8	190	816	96
3	10	160	1061	134
4	2	30	219	39
5	11	200	1201	111
6	8	110	772	52
<b>Total</b>	<b>50</b>	<b>860</b>	<b>5109</b>	<b>483</b>

Tabel 8. Detail solusi TS untuk 50 pelanggan

No Rute	Urutan Perjalanan
1	0 - 20 - 24 - 25 - 27 - 29 - 30 - 28 - 26 - 23 - 22 - 21 - 0
2	0 - 13 - 17 - 18 - 19 - 15 - 16 - 14 - 12 - 0
3	0 - 32 - 41 - 40 - 44 - 46 - 45 - 48 - 4 - 2 - 1 - 0
4	0 - 43 - 42 - 0
5	0 - 33 - 31 - 35 - 37 - 38 - 39 - 36 - 34 - 50 - 49 - 47 - 0
6	0 - 5 - 3 - 7 - 8 - 10 - 11 - 9 - 6 - 0

## SIMPULAN

Dari hasil penelitian penerapan *Genetic Algorithm* (GA), *Multiple Ant Colony System* (MACS) dan *Tabu Search* (TS) pada penyelesaian *Vehicle Routing Problem With Time Windows* (VRPTW) untuk 50 pelanggan menunjukkan bahwa kinerja dari algoritma MACS adalah yang terbaik, kemudian diikuti solusi dari algoritma TS dan terakhir adalah solusi GA. Penilaian dalam hal optimasi penentuan jumlah armada, total waktu perjalanan (*travel time*) dan total jarak perjalanan (*travel distance*).

## DAFTAR RUJUKAN

- [1] Andrea Roli, Christian Blum dan Marco Dorigo, *ACO for Maximal Constraint Satisfaction Problems*, MIC'2001 -- 4 th

Metaheuristics International Conference, Porto, Portugal, Juli 16-20, 2001

- [2] Dorigo, Marco dan Stützle, Thomas, 2004, *Ant Colony Optimization*, A Bradford Book, England.
- [3] Gambardella, Luca Maria, Taillard, Eric and Agazzi, Giovanni, 1999, *MACS-VRPTW Multiple Ant Colony System for Vehicle Routing Problems With Time Windows*, Lugano, Switzerland.

- [4] M. Dorigo. Optimization, Learning and Natural Algorithms (in Italian). PhD thesis. DEI, Politecnico di Milano, Italy, 1992. pp. 140.
- [5] M. Dorigo and G. Di Caro. The Ant Colony Optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 11-32. McGraw-Hill, 1999.
- [6] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Art. Life*, 5(2):137 (172), 1999.